

1. Write a servlet application for accepting hobbies of student through a web page using check boxes and display selected hobbies with some details about it.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Assignment3_Q1</title>
</head>
<body>
  <form action="Assignment3Q1" method="get">
    <label>Hobbies:</label><br>
    <input type="checkbox" name="hobbie" value="Playing" /> Playing<br><br>
    <input type="checkbox" name="hobbie" value="Singing" /> Singing<br><br>
    <input type="checkbox" name="hobbie" value="Listening" />
    Listening<br><br>
    <input type="checkbox" name="hobbie" value="Reading" /> Reading<br><br>
    <input type="submit" value="submit">
  </form>
</body>
</html>
```

Assignment3Q1.java

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class Assignment3Q1 extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse response)
        throws IOException, ServletException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String hobbies[] = req.getParameterValues("hobbie");

        if(hobbies != null) {
            out.println("Hobbies are : " );

            for(String hobb : hobbies) {

                if(hobb.equals("Playing")) {
                    out.println("<li>"+hobb+"</li>");
                    out.println("I like Playing");
                }
            }
        }
    }
}
```

```
    }
    if(hobb.equals("Listening")) {
        out.println("<li>" + hobb + "</li>");
        out.println("I like Listening Music");
    }
    if(hobb.equals("Singing")) {
        out.println("<li>" + hobb + "</li>");
        out.println("I like Singing ");
    }
    if(hobb.equals("Reading")) {
        out.println("<li>" + hobb + "</li>");
        out.println("I like reading Books");
    }
    }
    }
    else {
        out.println("please select the hobby");
    }
    out.close();
}
```

Output :

Hobbies:

☒ Playing

☒ Singing

☒ Listening

☐ Reading

Hobbies are :

- Playing

I like Playing

- Singing

I like Singing

- Listening

I like Listening Music

2. Write java program to accept user name and password store it in file. Accept password if following conditions holds true using regex: a. Minimum 8 characters should be taken b. It must have atleast 1 digit c. It must have atleast 1 special Character d. It must have atleast 1 Capital case letter.

```
package com.Assignment3;
import java.util.*;
import java.util.regex.*;
import java.io.*;
class PasswordValidator
{
    public static void main(String ar[]) throws Exception
    {
        Scanner sc=new Scanner(System.in);
        String uname,pass;
        System.out.println("Enter User name:");
        uname=sc.next();
        System.out.println("Enter password:");
        pass=sc.next();

        FileWriter f = new FileWriter("D:\\Shubham\\eclipse\\Assignment
3\\src\\main\\webapp\\password.txt");

        Pattern p=Pattern.compile("((?=.*[0-9])(?=.*[a-z])(?=.*[A-
Z])(?=.*[@%#$]).{8,20})");
        Matcher m=p.matcher(pass);
        if(m.matches())
        {
            f.write(uname+"\n"+pass);
            f.flush();
            System.out.println("Username and Password saved
successfully");
        }
        else
        {
            System.out.println("Password "+ pass +" is invalid");
        }
    }
}
```

Output :

```
Enter User name:
Admin
Enter password:
123456
Password 123456 is invalid
```

Roll No - 120

Div -B

Name - Shubham Kailas Deshmukh

Enter User name:

Shubham

Enter password:

Dshubham@17

Username and Password saved successfully

3. Write a JSP program for performing following operations on click of buttons (Take suitable input from user).

- a. Find square of entered number.**
- b. Check whether entered number is prime or not**
- c. Display current date.**
- d. Check whether entered number is Even number or odd number**

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Insert title here</title>
</head>

<body>
  <form action="Operation.jsp" method="get">
    Enter a Number : <input type="text" name="num"><br><br>
    <input type="radio" name="op" value="square">Find the Square of the
Number<br><br>
    <input type="radio" name="op" value="prime">Find given Number is Prime or
Not<br><br>
    <input type="radio" name="op" value="cdate">Find the Current Date<br><br>
    <input type="radio" name="op" value="even">Find the Number is Odd or
Even<br><br>
    <input type="submit" value="Submit">
  </form>
</body>

</html>
```

Operation.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
<%@ page import = "java.io.*,java.util.*" %>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Operations</title>
</head>

<body>
  <%
    int n = Integer.parseInt(request.getParameter("num"));
    String option = request.getParameter("op");
    int i;
```

```
if(option.equals("square")){
    int s = n*n;
    out.println("Square of Number : "+s);
}
if(option.equals("prime")){
    int f=0;
    for(i=2;i<=n/2;i++){
        if(n%i==0){
            f=1;
            break;
        }
    }
    if(f==0){
        out.println(n+ " is Prime Number");
    }
    else{
        out.println(n+ " is not Prime Number");
    }
}
if(option.equals("cdate")){
    Date d = new Date();
    out.println(d.toString());
}
if(option.equals("even")){
    if(n%2==0){
        out.println(n+ " is Even Number");
    }
    else{
        out.println(n+ " is Odd Number");
    }
}
%>
</body>
</html>
```

Output :

Enter a Number :

☐ Find the Square of the Number

☐ Find given Number is Prime or Not

☐ Find the Current Date

☐ Find the Number is Odd or Even

4. Write a program to demonstrate inter communication between thread using Banking domain.

```
package com.Assignment3;

class Customer {

    int amount = 10000;

    synchronized void withdraw(int amount) {
        System.out.println("Going to Withdraw..");
        if (this.amount < amount) {
            System.out.println("Less Balance , waiting for deposit");
            try {
                wait();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        this.amount = this.amount - amount;
        System.out.println("Withdrawal Completed");
    }

    synchronized void deposit(int amount) {
        System.out.println("Going to Deposit");
        this.amount = this.amount + amount;
        System.out.println("Deposit Completed");
        notify();
    }
}

public class Assignment3Q4 {

    public static void main(String[] args) {
        Customer c = new Customer();

        new Thread() {
            public void run() {
                c.withdraw(15000);
            }
        }
        .start();
        new Thread() {
            public void run() {
                c.deposit(10000);
            }
        }
        .start();
    }
}
```

Roll No - 120

Div -B

Name - Shubham Kailas Deshmukh

Output :

Going to Withdraw..

Less Balance , waiting for deposit

Going to Deposit

Deposit Completed

Withdrawal Completed

5. Write a Java program to design a employee registration form (employee id, age, first name,last name,contact number,address,email-id,department and salary)and insert all the record in a database.

```
<!DOCTYPE html>
<html>
<head>
    <title>Employee Registration Form</title>
</head>
<body>
    <h1>Employee Registration Form</h1>

    <form action="registerEmployee.jsp" method="post">
        <label for="id">Employee ID:</label>
        <input type="number" id="id" name="id" required><br><br>

        <label for="age">Age:</label>
        <input type="number" id="age" name="age" required><br><br>

        <label for="firstName">First Name:</label>
        <input type="text" id="firstName" name="firstName"
required><br><br>

        <label for="lastName">Last Name:</label>
        <input type="text" id="lastName" name="lastName"
required><br><br>

        <label for="contactNumber">Contact Number:</label>
        <input type="text" id="contactNumber" name="contactNumber"
required><br><br>

        <label for="address">Address:</label>
        <input type="text" id="address" name="address" required><br><br>

        <label for="emailId">Email ID:</label>
        <input type="email" id="emailId" name="emailId" required><br><br>

        <label for="department">Department:</label>
        <input type="text" id="department" name="department"
required><br><br>

        <label for="salary">Salary:</label>
        <input type="number" id="salary" name="salary" required><br><br>

        <input type="submit" value="Submit">
    </form>

</body>
```

</html>

registerEmployee.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import = "java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
    <title>Employee Registration Form - Results</title>
</head>
<body>
    <%
        int id = Integer.parseInt(request.getParameter("id"));
        int age = Integer.parseInt(request.getParameter("age"));
        String firstName = request.getParameter("firstName");
        String lastName = request.getParameter("lastName");
        String contactNumber = request.getParameter("contactNumber");
        String address = request.getParameter("address");
        String emailId = request.getParameter("emailId");
        String department = request.getParameter("department");
        double salary = Double.parseDouble(request.getParameter("salary"));
        Connection con;
        PreparedStatement ps;
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mca", "root", "");
            String sql = "insert into emp values (?, ?, ?, ?, ?, ?, ?, ?)";
            ps = con.prepareStatement(sql);
            ps.setInt(1, id);
            ps.setInt(2, age);
            ps.setString(3, firstName);
            ps.setString(4, lastName);
            ps.setString(5, contactNumber);
            ps.setString(6, address);
            ps.setString(7, emailId);
            ps.setString(8, department);
            ps.setDouble(9, salary);
            int res = ps.executeUpdate();
            out.print(res + " Employee registered Successfully");

        } catch (Exception e) {
            e.printStackTrace();
        }
    %>
</body>
```

Roll No - 120

Div -B

Name - Shubham Kailas Deshmukh

</html>

Output :

Employee Registration Form

Employee ID:

Age:

First Name:

Last Name:

Contact Number:

Address:

Email ID:

Department:

Salary:

6. Create a hash table program and perform the following operations on it like**hash(),set(),get(),remove(),display()**

```
package com.Assignment3;

import java.util.ArrayList;
import java.util.LinkedList;

import java.util.HashMap;

public class HashTable {
    private HashMap<String, String> hashtable;

    public HashTable() {
        hashtable = new HashMap<>();
    }

    // Calculate the hash code for a given key
    public int hash(String key) {
        return key.hashCode() % hashtable.size();
    }

    // Add a key-value pair to the hash table
    public void set(String key, String value) {
        hashtable.put(key, value);
    }

    // Get the value associated with a key in the hash table
    public String get(String key) {
        return hashtable.get(key);
    }

    // Remove a key-value pair from the hash table
    public void remove(String key) {
        hashtable.remove(key);
    }

    // Display the contents of the hash table
    public void display() {
        System.out.println(hashtable);
    }

    public static void main(String[] args) {
        HashTable hashTable = new HashTable();

        // Add some key-value pairs to the hash table
        hashTable.set("fruit", "banana");
        hashTable.set("movie", "Dhoom");
        hashTable.set("pen", "Black");
    }
}
```

```
System.out.println("Whole HashTable : ");
hashTable.display();

System.out.println("Get Method : ");
System.out.println(hashTable.get("movie"));

System.out.println("Remove Method : ");
hashTable.remove("pen");

System.out.println("Display Hashtable after deletion");
hashTable.display();
}
}
```

Output :

```
Whole HashTable :
{movie=Dhoom, fruit=banana, pen=Black}
Get Method :
Dhoom
Remove Method :
Display Hashtable after deletion
{movie=Dhoom, fruit=banana}
```

7.Create a program to perform divide and conquer binary search algorithm by iterative approach.

```
let data = [5, 9, 13, 17, 45, 67, 89, 100];
let find = 101;
let start = 0;
let end = data.length - 1;
let position = undefined;
while (start <= end) {
  let mid = Math.floor((start + end) / 2);
  if (data[mid] === find) {
    position = mid;
    break;
  } else if (data[mid] < find) {
    start = mid + 1;
  } else {
    end = mid - 1;
  }
}
console.log("Element found at : "+position);
```

Output :

Element found at : 7

8.Create a program to perform divide and conquer binary search algorithm by recursive approach.

```
function recursiveFunction(arr, x, start, end) {  
  // Base Condition  
  if (start > end) return false;  
  let mid = Math.floor((start + end) / 2);  
  if (arr[mid] === x) return true;  
  if (arr[mid] > x) return recursiveFunction(arr, x, start, mid - 1);  
  else return recursiveFunction(arr, x, mid + 1, end);  
}  
let arr = [1, 3, 5, 7, 8, 9];  
let x = 5;  
if (recursiveFunction(arr, x, 0, arr.length - 1)) {  
  console.log("Element found!");  
}  
else {  
  console.log("Element not found!");  
}  
if (recursiveFunction(arr, x, 0, arr.length - 1)) {  
  console.log("Element found");  
} else {  
  console.log("Element not found");  
}  
}
```

Output :

Element found!

Element found

9. Create a program to perform largest sum of sub array using kandel's algorithm.

```
class Kandealgo {  
  constructor() {  
  }  
  maxSumSubArray(inputarray) {  
    let maximum_sum = 0;  
    let current_sum = 0;  
    for (let i = 0; i < inputarray.length; i++) {  
      current_sum = current_sum + inputarray[i];  
      if (current_sum > maximum_sum) {  
        maximum_sum = current_sum;  
      } if (current_sum < 0) {  
        current_sum = 0;  
      }  
    }  
    return maximum_sum;  
  }  
}  
const op = new Kandealgo();  
const inputarray = [5, -4, -2, 6, -1];  
const o = op.maxSumSubArray(inputarray);  
console.log(o);
```

Output :

6

10. Create a program to perform tower of Hanoi.

```
function towerOfHanoi(n, source, destination, auxiliary) {  
  if (n === 1) {  
    console.log(`Move disk 1 from ${source} to ${destination}`);  
    return;  
  }  
  towerOfHanoi(n - 1, source, auxiliary, destination);  
  console.log(`Move disk ${n} from ${source} to ${destination}`);  
  towerOfHanoi(n - 1, auxiliary, destination, source);  
}  
  
// example usage  
towerOfHanoi(3, 'A', 'C', 'B');
```

Output :

```
Move disk 1 from A to C  
Move disk 2 from A to B  
Move disk 1 from C to B  
Move disk 3 from A to C  
Move disk 1 from B to A  
Move disk 2 from B to C  
Move disk 1 from A to C
```